

④

(19)日本国特許庁(J P)

(12) 公開特許公報(A)

(11)特許出願公開番号

特開平5-189243

(43)公開日 平成5年(1993)7月30日

(51)Int.Cl.⁶

G 0 6 F 9/45

識別記号

庁内整理番号

9292-5B

F I

G 0 6 F 9/44

3 2 2 F

技術表示箇所

審査請求 未請求 請求項の数1(全7頁)

(21)出願番号 特願平4-4869

(22)出願日 平成4年(1992)1月14日

(71)出願人 000005223

富士通株式会社

神奈川県川崎市中原区上小田中1015番地

(72)発明者 堀田 耕一郎

神奈川県川崎市中原区上小田中1015番地

富士通株式会社内

(72)発明者 中平 直司

神奈川県川崎市中原区上小田中1015番地

富士通株式会社内

(74)代理人 弁理士 小笠原 哲義 (外2名)

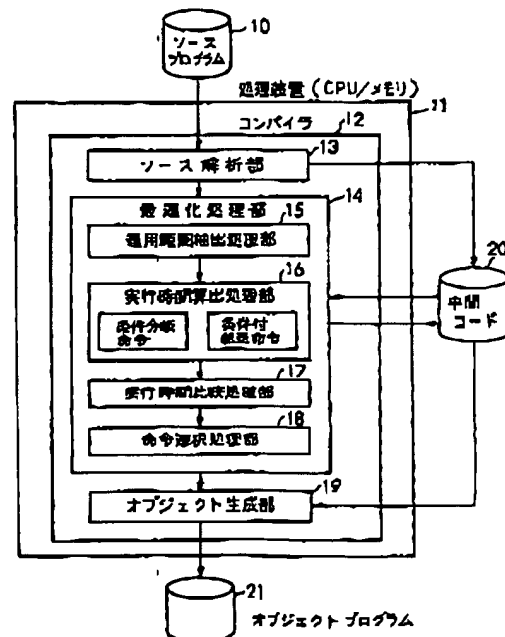
(54)【発明の名称】 条件付演算のコンパイル処理装置

(57)【要約】

【目的】本発明は、条件付演算を含むソースプログラム10をオブジェクトプログラム21に翻訳する条件付演算のコンパイル処理装置に関し、条件の成立する確率に応じて、最適な命令列を選択することにより、条件付演算の高速化を実現することを目的とする。

【構成】適用範囲抽出処理部15は、ソースプログラム10の解析結果から、一つの条件文によって制御を受ける範囲を抽出する。実行時間算出処理部16は、その適用範囲について、条件文の成立確率情報と条件によって支配される演算の実行時間の情報を使用し、条件分岐命令を用いた場合の実行時間と条件付転送命令を用いた場合の実行時間とを算出する。実行時間比較処理部17により、その算出した実行時間を比較し、命令選択処理部18により、条件分岐命令と条件付転送命令のうち、短い実行時間の命令列が翻訳結果となるように命令列を選択する。

本発明の構成例



【特許請求の範囲】

【請求項1】 条件文を含む高級言語で記述されたプログラム(10)を、条件付転送命令の実行機構を持つ計算機で実行可能なオブジェクトプログラム(21)に翻訳するコンパイル処理装置において、翻訳対象のプログラムまたはその解析結果から、一つの条件文によって制御を受ける範囲を抽出する適用範囲抽出処理手段(15)と、抽出した適用範囲について、条件文の成立確率情報と条件によって支配される演算の実行時間の情報を使用し、条件分岐命令を用いた場合の翻訳結果の実行時間と条件付転送命令を用いた場合の翻訳結果の実行時間とを算出する実行時間算出処理手段(16)と、算出した実行時間を比較する実行時間比較処理手段(17)と、実行時間の比較により、条件分岐命令を用いた命令列と条件付転送命令を用いた命令列のうち、短い実行時間の命令列が翻訳結果となるように選択する命令選択処理手段(18)とを備えたことを特徴とする条件付演算のコンパイル処理装置。

【発明の詳細な説明】

【0001】

【産業上の利用分野】本発明は、条件付演算を含むソースプログラムを、最適化により高速に実行できるオブジェクトプログラムに翻訳する条件付演算のコンパイル処理装置に関する。

【0002】FORTRANやその他のプログラムでは、ある条件が成立したときに、所定の演算を実行するような条件付演算が多く用いられる。このような条件付演算の実行を高速化することにより、プログラムの実行性能を上げることができる。

【0003】

【従来の技術】図3は従来技術の説明図である。図3に示すソースプログラム10は、IF文で記述された条件式が成立したときに、『式』を計算し、結果をAの領域に代入するプログラムである。このように、特定の条件式が成立したとき、または成立しなかったときにだけ実行する演算を、条件付演算という。

【0004】従来、図3に示すような条件付演算を含むソースプログラム10をオブジェクトプログラムに翻訳する場合、演算およびその結果を目的の領域に代入する命令を、条件付分岐命令で制御することによって実現していた。

【0005】ところで、最近では計算機の命令に条件付転送命令を備えたものが用いられ始めている。条件付転送命令は、1つの命令でオペランドで指定された条件が成立したときにのみ、転送(代入)を実行する命令である。このような条件付転送命令を備えた計算機で実行するプログラムを翻訳する場合、図3に示すようなソースプログラム10を、条件分岐命令ではなく、条件付転送命令を用いたものに翻訳することも可能である。

【0006】そこで、ソースプログラム10を翻訳する場合に、コンパイラでは、条件付転送命令が使用可であ

るか不可であることを判定して、使用可であれば、図3に示すような条件転送方式の擬似命令30b、使用不可であれば、条件分岐方式の擬似命令30aに展開することが行われていた。なお、使用可/不可の判定には、計算機が条件付転送命令を備えているかどうかだけでなく、例えば演算の際のオーバーフロー等の例外の発生を無視できるかどうかなどの外的条件も判断される。

【0007】なお、図3に示す条件分岐方式の擬似命令30aでは、条件式の値をt1に代入し、それが『偽』であれば、addrのアドレスに分岐する。t1が『真』であれば、Aに代入する式の演算を行い、結果をAに代入するようになっている。

【0008】一方、条件転送方式の擬似命令30bでは、条件式の値をt1に代入し、Aに代入する式の演算は無条件に行い、条件式の値t1が『真』のときだけ条件付転送命令によって、Aにt2の演算結果を代入している。

【0009】

【発明が解決しようとする課題】一般に、計算機において分岐命令のオーバーヘッドは大きく、条件分岐命令を使用した方式は、実際に分岐を行う場合には、演算の実行時間を省くことができるというメリットはあるものの、分岐を行わない場合に分岐命令のオーバーヘッドが負担になる。

【0010】一方、条件付転送命令を使用した場合には、実際に結果の転送が不必要な場合にも式の演算を実行するため、上記の条件分岐方式でいう「分岐を行う場合」と比較して、式を演算する分のオーバーヘッドが負担となる。

【0011】以上のように、条件分岐方式および条件転送方式には、それぞれ一長一短があり、条件の内容によってどちらが最適であるかが変わってくるが、従来考えられている技術では、どちらの方式によるオブジェクトの展開を行うかが、一律に決められるため、最善なオブジェクトプログラムが生成されるとは限らなかった。

【0012】本発明は上記問題点の解決を図り、条件の成立する確率に応じて、最適な命令列を選択することにより、条件付演算の高速化を実現することを目的としている。

【0013】

【課題を解決するための手段】図1は本発明の構成例を示すブロック図である。ソースプログラム10は、例えばFORTRANやC言語等の高級言語で記述された翻訳対象のプログラムである。処理装置11は、CPUやメモリなどからなる装置である。コンパイラ12は、ソースプログラム10をターゲットの計算機で実行可能なオブジェクトプログラム21に翻訳するプログラムである。

【0014】ソース解析部13は、ソースプログラム10を入力し、構文解析および意味解析を行って、解析結

果からオブジェクトプログラム21のもととなる中間コード20を生成し出力する処理手段である。最適化処理部14は、中間コード20について実行高速化のための処理を行うものである。オブジェクト生成部19は、最適化された中間コード20をもとに、領域の割り付けなどを行い、機械語レベルのオブジェクトプログラム21を生成し出力する処理手段である。

【0015】本発明では、最適化処理部14の部分に、適用範囲抽出処理部15、実行時間算出処理部16、実行時間比較処理部17、命令選択処理部18を有する。適用範囲抽出処理部15は、ソースプログラム10を解析した中間コード20から、一つの条件文によって制御を受ける範囲を抽出する処理手段である。実行時間算出処理部16は、適用範囲抽出処理部15により抽出した適用範囲について、条件文の成立確率情報と条件によって支配される演算の実行時間の情報を使用し、条件分岐命令を用いた場合の翻訳結果の実行時間と条件付転送命令を用いた場合の翻訳結果の実行時間とを算出する処理手段である。実行時間比較処理部17は、実行時間算出処理部16が算出した実行時間を比較する処理手段である。命令選択処理部18は、実行時間の比較により、条件分岐命令を用いた命令列と条件付転送命令を用いた命令列のうち、短い実行時間の命令列が翻訳結果となるように選択する処理手段である。

【0016】

【作用】ソースプログラム10の条件文を機械語に展開する場合、条件分岐命令を用いたほうがよいか、条件付転送命令を用いたほうがよいかは、条件の成立確率および条件によって支配される演算の実行時間がどれくらいになるかによって変わってくる。そこで、本発明では、条件文の条件の成立確率と条件によって制御される演算の単体実行時間のデータをもとに、条件付演算の実行時間の期待値を算出し、最適な命令列を選択する。

【0017】

【実施例】本発明の実施例を説明するに先立ち、本実施例で用いる条件付転送命令の仕様について説明する。

【0018】条件付転送命令は、次のような命令構成になっている。

『 move [x] r1, r2, cc 』

これは、第3オペランドccの内容が[x]部分で定まる命令コードの条件に合致した場合に、第2オペランドr2の内容を第1オペランドr1に転送する命令である。条件は命令コードによって選択することが可能であり、例えば「2つのオペランドの比較の結果、値が一致した場合」というときには、「move eq r1, r2, cc」のような命令になる。

【0019】ccはコンディションコードを保持するレジスタの番号を示す。ccの内容は、この命令に先行する命令によって設定される。一般には、比較命令によって設定されることが考えられるが、一般の演算命令によ

っても演算結果に応じて設定されることもある。この種のレジスタが1個だけしか存在しないハードウェアの場合には、第3オペランドは不要である。

【0020】例えば、変数Aの内容が変数Bの内容よりも大きい場合に、変数Bに変数Aの内容を代入するというときの擬似命令列は、以下ようになる。なお、擬似命令とは、実際には2進の機械語命令を記号で表したものである。

【0021】

```
load      r1, A
load      r2, B
comp      cc, r1, r2
movegt    r2, r1, cc
store     r2, B
```

ここで、loadは、第2オペランドの内容を第1オペランドで指定されたレジスタにロードする命令、compは、第2オペランドの内容と第3オペランドの内容を比較して指定されたコンディションコードを設定する命令、movegtは、コンディションコードの内容が『より大きい』を示しているときのみ、第2オペランドの内容を第1オペランドで示されたレジスタに転送する命令、storeは、第1オペランドで指定されたレジスタの内容を第2オペランドで指定されたメモリに書き込む命令である。

【0022】図2は本発明の実施例説明図である。以下、図2の(イ)に示すソースプログラム10を翻訳する場合を例に、図1に示す最適化処理部14の処理について説明する。他の処理については従来と同様である。

【0023】(1)適用範囲抽出処理部15は、本方式を適用する範囲を抽出する。本実施例では、図2の(イ)に示すようなソースプログラム10を、一旦、ソース解析部13により、図2の(ロ)に示すような条件分岐方式による中間コード20aに変換し、適用範囲は、その中間コード20aから、以下の条件によって求める。

【0024】① 条件分岐を示す中間コードとその分岐先との間であること、かつ、対象範囲に他から分岐してくるパスがないこと、かつ、対象範囲から外へ分岐することがないこと。

【0025】これらの条件のどれかが満たされない場合には、適用範囲外であり、中間コード20aについての最適化は行わない。これらの条件がすべて満たされた場合、条件分岐方式と条件転送方式のどちらで命令を展開するかを検査を行う。図2に示す例では、これらの条件が満たされたものとする。

【0026】(2)次に、実行時間算出処理部16により、処理(1)で抽出した範囲を調べて、図2の(ロ)に示すような条件分岐方式で命令を展開した場合と、図2の(ハ)に示すような条件転送方式で命令を展開した場合の実行時間を算出する。

【0027】ここでは、条件分岐命令による分岐時間が

(4)

特開平5-189243

分岐する場合に**b**、分岐しない場合に**n**であったとする。『式』の演算時間は**e**であるが、条件分岐方式の場合、分岐条件が成立したときには実際の演算時間が0となる。

【0028】『式』の結果をAに代入する一般の転送命令の実行時間は**t**であり、条件付転送命令による実行時間は**t'**であるが、この差は非常に小さい。一方、条件分岐の時間である**b**および**n**は、転送命令の時間**t**に比べて非常に大きく、昨今のRISC方式と呼ばれる計算機アーキテクチャでは、数十倍の開きがあることもある。

【0029】図2の(ロ)に示すように、条件分岐方式により命令を展開した場合、その全実行時間**Tb**は、分岐条件不成立時に**Tb = b**であり、分岐条件成立時に**Tb = n + e + t**である。分岐の条件式が真になる確率、すなわち分岐が起こらない確率を

としたときに、条件分岐方式の平均実行時間は、

$$Tb = p \cdot (n + e + t) + (1 - p) b$$

で求められる。

【0030】一方、図2の(ハ)に示すように、条件転送方式により命令を展開した場合、その実行時間**Tt**は、常に、**Tt = e + t'**となる。演算時間などの各実行時間は、中間コードの種別とターゲットの計算機によって決まる命令の実行時間とのデータから算出することができる。例えば、コンパイラ12内に、あらかじめ各中間コードまたは擬似命令ごとに実行時間が登録されたテーブルを用意し、そのデータを参照すればよい。

【0031】なお、図2の(ハ)に示す例では、条件付転送命令が1回しか用いられていないが、実際には適用範囲内で代入された値のうち、範囲外で参照される値が複数あれば、条件付転送命令を複数回使用する必要がある。そこで、適用範囲内での値の代入のうち、範囲外でその値が参照されるものの数を調べる。範囲外で値が参照されるかどうかは、一般に知られている「データフロー情報」によって検出可能である。これによって、条件転送方式の場合の条件付転送命令が必要となる数が求まり、1個当りの単純転送命令と条件付転送命令との実行時間の差(**t' - t**)を考慮し、実行時間**Tt = e + t'**を求めることができる。

【0032】条件分岐方式の平均実行時間の計算で必要となる条件文の分岐確率は、例えば翻訳時制御行のようなソースプログラムに与えられる指示によって決めてもよいし、または、一般に知られているプログラムのチューニングツールにより、実際にテスト走行させ、どのルートを何回通ったかというようなチューニング出力情報を自動的に読み込んで、決定するようにしてもよい。

【0033】(3) 図2の(ロ)に示す条件分岐方式の実行時間**Tb**と、図2の(ハ)に示す条件転送方式の実行時間**Tt**を算出したならば、実行時間比較処理部17によりその大小を比較する。

【0034】(4) 命令選択処理部18は、条件転送方式のほうが実行時間が短ければ、図2の(ロ)に示す中間コード20aを、図2の(ハ)に示す中間コード20bに変更する。具体的には、**Tb > Tt**ならば、条件分岐の中間コード(Bif...)を削除し、対象範囲内において範囲外から値が参照される代入を、条件付転送命令用の中間コードに置換する。**Tb ≤ Tt**ならば、図2の(ロ)に示す中間コード20aをそのままオブジェクト生成部19へ引き渡す。

【0035】以上の処理をプログラムの全範囲に対して繰り返すことにより、条件付演算について高速実行が可能な命令列が生成されることになる。

【0036】

【発明の効果】特に、小規模で、複数の命令を並列に実行することによって高速化を図る計算機の場合、分岐命令は命令の並列度を落とすため、性能を低下させる原因になる。そのため、条件付転送命令の使用は、高速化のために必須と言える。しかし、すべての条件付演算に条件付転送命令を用いると、条件文の分岐確率によって、かえって性能が低下することもある。本発明によれば、分岐確率を考慮し、条件分岐方式と条件転送方式のうち、実行効率のよいほうを自動的に選択して、その命令列に展開するので、翻訳結果の実行性能を高めることができる。

【図面の簡単な説明】

【図1】本発明の構成例を示すブロック図である。

【図2】本発明の実施例説明図である。

【図3】従来技術の説明図である。

【符号の説明】

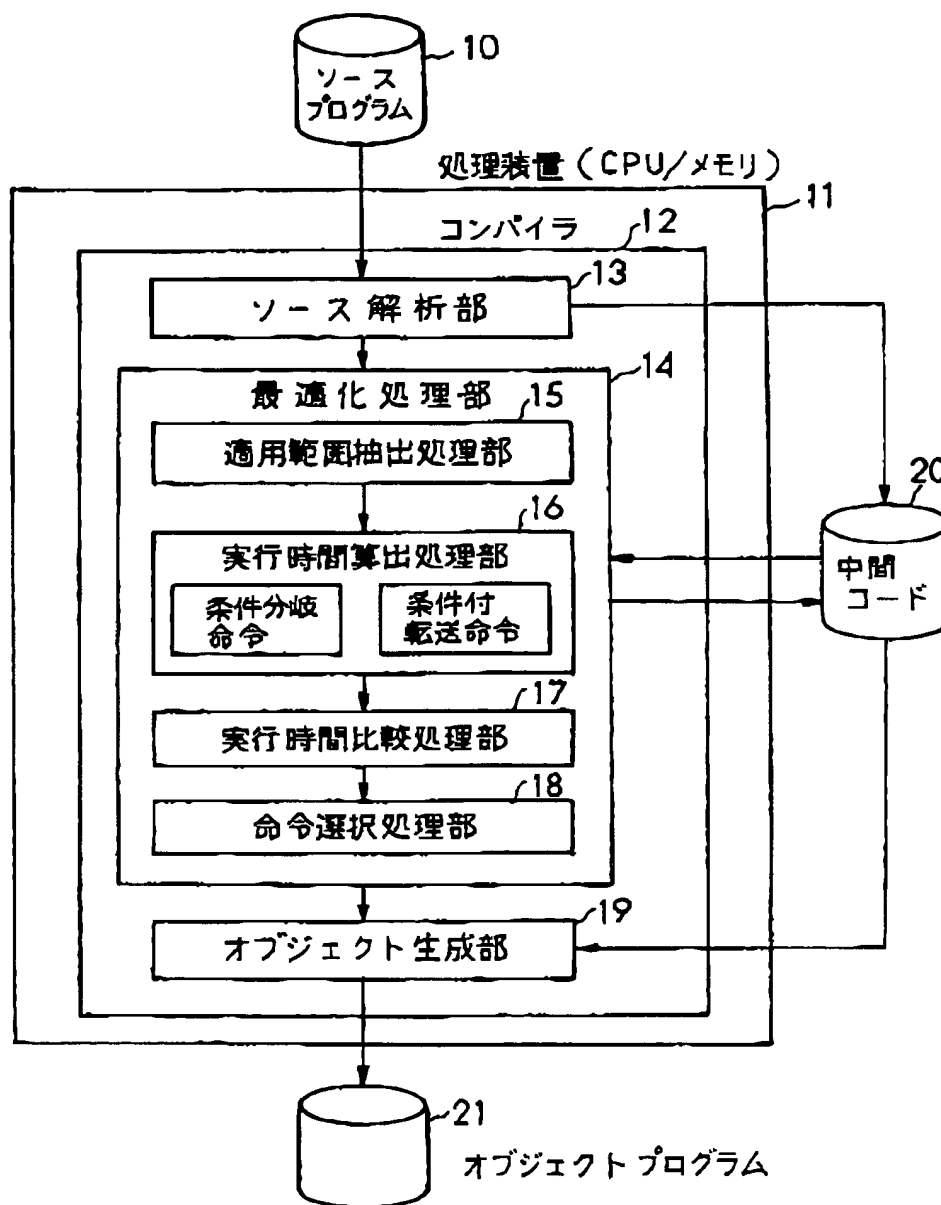
- 10 ソースプログラム
- 11 処理装置
- 12 コンパイラ
- 13 ソース解析部
- 14 最適化処理部
- 15 適用範囲抽出処理部
- 16 実行時間算出処理部
- 17 実行時間比較処理部
- 18 命令選択処理部
- 19 オブジェクト生成部
- 20 中間コード
- 21 オブジェクトプログラム

(5)

特開平5-189243

【図1】

本発明の構成例

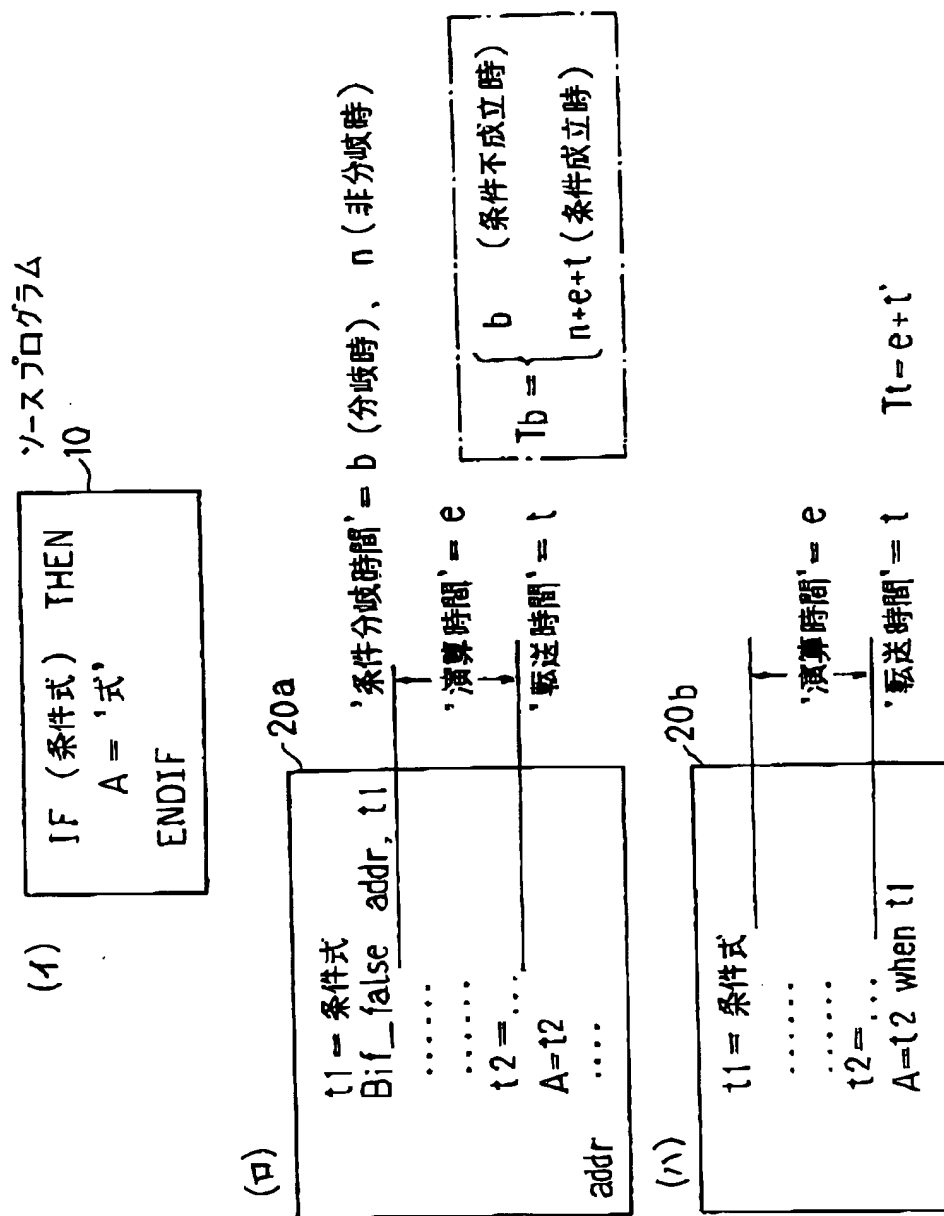


(6)

特開平5-189243

【図2】

本発明の実施例説明図



(7)

特開平5-189243

【図3】

従来技術の説明図

